Explicit Motion Disentangling for Efficient Optical Flow Estimation

Changxing Deng[1]*, Ao Luo[2]*, Haibin Huang[3], Shaodan Ma[1], Jiangyu Liu[2], Shuaicheng Liu[4,2]†
[1]University of Macau [2]Megvii Technology [3]Kuaishou Technology
[4]University of Electronic Science and Technology of China

Abstract

In this paper, we propose a novel framework for optical flow estimation that achieves a good balance between performance and efficiency. Our approach involves disentangling global motion learning from local flow estimation, treating global matching and local refinement as separate stages. We offer two key insights: First, the multi-scale 4D cost-volume based recurrent flow decoder is computationally expensive and unnecessary for handling small displacement. With the separation, we can utilize lightweight methods for both parts and maintain similar performance. Second, a dense and robust global matching is essential for both flow initialization as well as stable and fast convergence for the refinement stage. Towards this end, we introduce EMD-Flow, a framework that explicitly separates global motion estimation from the recurrent refinement stage. We propose two novel modules: Multi-scale Motion Aggregation (MMA) and Confidence-induced Flow Propagation (CFP). These modules leverage cross-scale matching prior and self-contained confidence maps to handle the ambiguities of dense matching in a global manner, generating a dense initial flow. Additionally, a lightweight decoding module is followed to handle small displacements, resulting in an efficient yet robust flow estimation framework. We further conduct comprehensive experiments on standard optical flow benchmarks with the proposed framework, and the experimental results demonstrate its superior balance between performance and runtime. Code is available at `https://github.com/gddcx/EMD-Flow`.

1. Introduction

Optical flow represents the 2D motion field between successive video frames. It is a fundamental computer vision task and has various downstream applications, *e.g.* video frame interpolation [13], video super-resolution [8], visual tracking [33] and motion detection [26]. Different from traditional energy-based or matching-based optimizations, deep learning based methods [30, 15] have made great progress by introducing the cost-volume based regression paradigm in a pyramid structure. Recently, the recurrent regression framework [32] has become a mainstream approach for optical flow, and advanced methods like attention-based operations [18, 21, 38], graph models [23], and latent cost-volume augmentation [12] have been developed to improve its performance. However, these methods often require extra computational resources and consume significant inference time, limiting their application in real-world scenarios. Therefore, a critical question arises: can we improve the accuracy of flow estimation while maintaining high runtime efficiency?

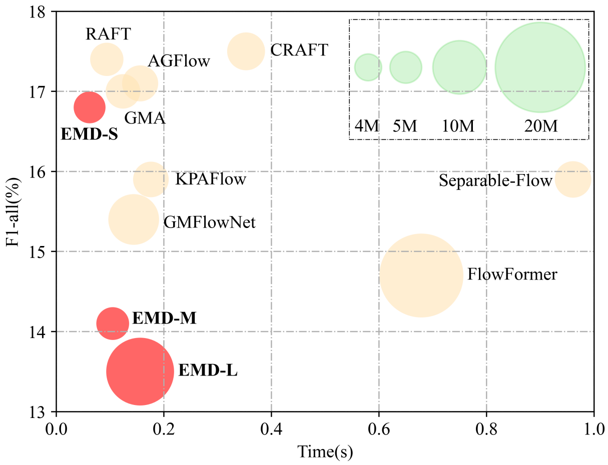


Figure 1: **Comparison with state-of-the-art methods in terms of inference accuracy (F1-all), runtime (s) and model size (M).** All models are trained on "C + T", and evaluated on KITTI-15 [25] (image size 375×1242) with a single NVIDIA A100 card. Our EMD-Flow models not only demonstrate substantial performance enhancements compared to state-of-the-art methods but also achieve significant reductions in computational overhead.

To understand the trade-off between accuracy and runtime efficiency, we empirically analyze the recurrent pre-

*Equal contribution. †Corresponding Author.

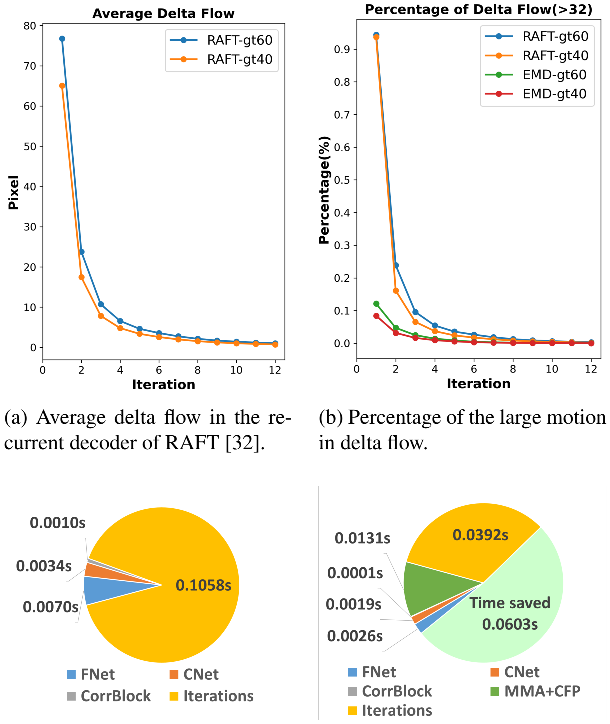
diction paradigm in the RAFT model [32] and its computational overhead. The core component of RAFT is the iterative decoder, which obtains the final prediction by iteratively refining the flow estimate. The statistical analysis reveals that the early iterations primarily address the challenge of handling large displacements, while the subsequent iterations concentrate on small-scale motion and local refinement, as shown in Fig. 2a and Fig. 2b. In terms of computational overhead shown in Fig. 2c, we find that all iterations take up about 90% of the running time of the model, with the corresponding parameter ratio of 58.5%.

Based on our analysis, we draw the following conclusions: **i**) RAFT primarily handles large motion in the early iterations and small motion in the later iterations. **ii**) When all points are estimated from the same starting point, the iterative procedure becomes out of sync because large displacement requires more iterations than small displacement. **iii**) The full recurrent unit based on multi-scale 4D cost-volume is computationally expensive and unnecessary for handling small displacement.

Towards this goal, we propose a novel flow network with an Explicit Motion Disentangling (EMD) strategy that effectively handles large motion while maintaining runtime efficiency. Our key insight is to disentangle the global motion learning process from the complex recurrent decoder and employ a lightweight decoding module to handle small displacements. Specifically, we introduce Confidence-induced Flow Propagation (CFP), a multi-scale and confidence-induced module that utilizes cross-scale matching priors, global context relations, and self-contained confidence maps to generate an accurate initial dense flow map. Additionally, we present Multi-scale Motion Aggregation (MMA), a feature enhancement and multi-scale feature matching module that aggregates mutual dependencies of features and utilizes cross-scale information for improved initial flow estimation.

Based on the proposed CFP and MMA modules, we develop an efficient and powerful optical flow estimation model, namely **EMD-Flow**. Benefitting from the property of CFP and MMA modules, our EMD-Flow is able to effectively handle large motion before recurrent scheme (see Fig. 2b). Moreover, our full network is designed with a high-efficiency principle to ensure a cost-effective model, as in Fig. 2d. We conduct comprehensive experiments to demonstrate that our approach achieves both high efficiency and excellent performance on the standard benchmarks, including Sintel and KITTI. To summarize, the main contributions of our work are as follows:

- We introduce Explicit Motion Disentangling (EMD) strategy to handle global motion and small displacement estimation separately, achieving a better performance-runtime balance.



(a) Average delta flow in the recurrent decoder of RAFT [32].

(b) Percentage of the large motion in delta flow.

(c) RAFT time distribution.

(d) EMD-Flow time distribution.

Figure 2: **Analyses of the delta flow and runtime comparison with RAFT[32]**. The average delta flow in (a) and (b) are counted on Sintel clean. "-gt40" and "-gt60" indicate the strength of ground truth flow at the sampled points are larger than 40 and 60 pixels, respectively. We regard the flow vectors with value > 32 pixels as the large motion.

- We propose Confidence-induced Flow Propagation (CFP) and Multi-scale Motion Aggregation (MMA) modules, which improve the accuracy of flow estimation while maintaining runtime efficiency.

- Our proposed model, EMD-Flow, achieves state-of-the-art performance on standard benchmarks while consuming fewer computational resources, demonstrating the effectiveness of our approach.

2. Related Work

Optical Flow Estimation. Optical flow is a long-standing task to estimate the 2D motion field among successive frames in the video. Traditionally, optical flow is regarded as an energy minimization problem [11, 2, 3, 6]. Although some better methods are proposed to enhance feature similarity and motion smoothness [4, 37, 27], limited by the handcrafted features, traditional methods are still hard to handle the cases in complex environments.

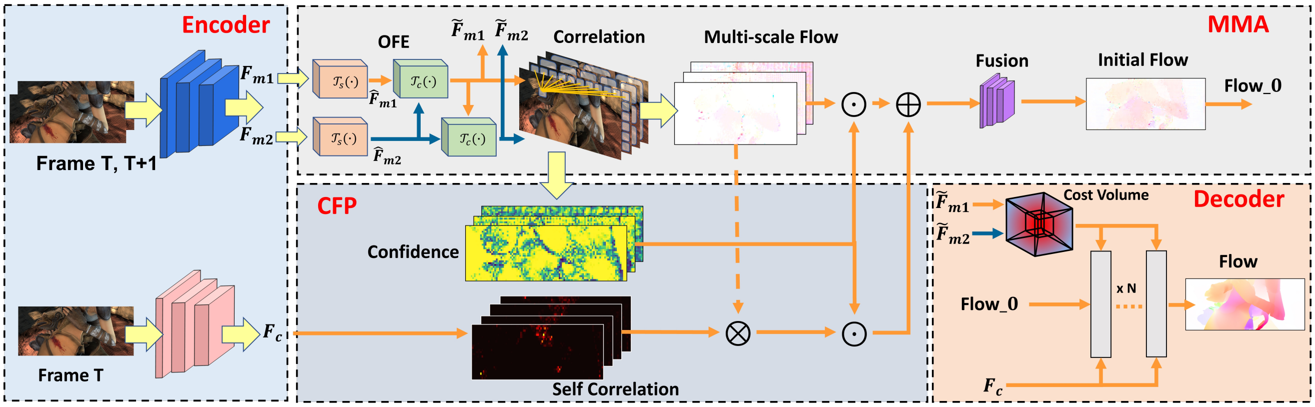


Figure 3: **The overall architecture of proposed EMD-Flow.** It has two core modules: **i**) The **M**ulti-scale **M**otion **A**ggregation (MMA) module, and **ii**) The **C**onfidence-induced **F**low **P**ropagation (CFP) module. ”⊗”, ”⊕” and ”⊙” denote multiplication, weighted sum and a selective strategy proposed in EMD-Flow, respectively.

The recent advances in deep learning bring a huge improvement in this field. FlowNet [9] presents the first end-to-end network to handle this task. Then a series of methods based on neural networks [16, 28, 30, 14, 20] are proposed to improve performance in a coarse-to-fine or iterative manner. Recently, RAFT [32] introduces a novel recurrent flow estimation framework based on multi-scale 4D cost volume, achieving great improvements on all benchmarks. Drawing upon the recurrent flow refinement scheme, numerous recent approaches [18, 10, 12, 22] have significantly enhanced the performance and reliability of optical flow predictions. [29] and [38] use attention mechanism to enhance the feature extracted from feature encoder before the construction of 4D cost volume. [18] learn to estimate the hidden motion by context similarity and [23, 21] takes a further step to utilize spatial relation by graph reasoning and kernel patch attention, respectively. [12] fully takes advantage of the attention mechanism to optimize the constructing and searching function of cost volume, greatly improving the performance on multiple datasets.

Although the improved methods based on RAFT can achieve better performance, the running time consumption also increases due to the additional modules. Unlike prior improved methods, we propose to use explicit global flow estimation to handle large displacement, which greatly reduces the iterations and parameters of the recurrent module of RAFT. This approach not only creates a lightweight flow network but also offers the convenience of enhancing flow refinement with an effective initial flow.

Explicit matching for optical flow. The classical energy-based flow estimation methods [11, 2, 3, 6] usually fail to handle the challenges of large motion. To remedy the problem, a matching step is introduced in [1, 5, 34] to find the corresponding pixel before energy-based optimization. However, limited by the handcrafted feature, the matching is inaccurate. In the deep learning era, explicit matching is applied in recent optical flow networks [38, 36], which helps to obtain remarkable performance. GMFlowNet [38] employs an argmax operation on 4D cost volume to perform a sparse global matching before the RAFT-like architecture. GMFlow [36] estimates global flow by weighting coordinates based on cross-similarities between two frames.

In contrast to existing approaches, we employ a multi-scale strategy to enhance accuracy in global matching. Furthermore, we leverage a self-contained confidence map to effectively handle uncertain regions and further improve precision. Benefitting from these advanced modules, our EMD-Flow exhibits a remarkable balance between performance and runtime, highlighting its superiority. To the best of our knowledge, we are pioneering in simultaneously enhancing flow precision (on both Sintel and KITTI datasets) and computational efficiency.

3. Approach

We propose a novel framework termed Explicit Motion Disentangling for Optical Flow (EMD-Flow) estimation. The architecture is shown in Fig. 3, which consists of two core components, *i.e.,* Multi-scale Motion Aggregation (MMA) and Confidence-induced Flow Propagation (CFP). Details are elaborated in the following sections.

3.1. Multi-scale Motion Aggregation

An overview of our approach is illustrated in Fig. 3. As can be seen, our MMA consists of two parts, *i.e.,* Orderly Feature Extraction (OFE) module based on Transformers and multi-scale flow fusion to obtain initial optical flow. Specifically, we design OFE module $\mathcal{T}(\cdot)$ to capture the long-range and cross-image features for representation enhancement, which contains spatial relation learning block

$\mathcal{T}_s(\cdot)$ and mutual dependencies learning block $\mathcal{T}_c(\cdot)$. Given the feature $F_{m1} \in \mathbb{R}^{h\times w\times d_m}$ and $F_{m2} \in \mathbb{R}^{h\times w\times d_m}$ extracted from feature encoder, the OFE can be defined as:

$$\begin{aligned}
\hat{F}_{m1} &= \mathcal{T}_s(F_{m1}),\\
\hat{F}_{m2} &= \mathcal{T}_s(F_{m2}),\\
\tilde{F}_{m1} &= \mathcal{T}_c(\hat{F}_{m1}, \hat{F}_{m2}),\\
\tilde{F}_{m2} &= \mathcal{T}_c(\hat{F}_{m2}, \tilde{F}_{m1}).
\end{aligned} \tag{1}$$

$\mathcal{T}_s(\cdot)$ contains self-attention and feed forward network. It is used to improve the spatial relation of features and avoid being ambiguous in the construction of cost volume. $\mathcal{T}_c(\cdot)$ denotes cross-attention block, which is utilized to deliver image-level mutual information.

Note that, unlike previous works [36, 31] applying $\mathcal{T}_c(\cdot)$ in a simply parallel way, we carefully design the operations with an orderly information passing structure. As shown in Fig. 3, we first deliver the enhanced feature $\hat{F}_{m2}$ to $\hat{F}_{m1}$ for cross-attention learning, and then the combined feature $\tilde{F}_{m1}$ is sent back to the next cross-attention with $\hat{F}_{m2}$. In this way, the feature $\hat{F}_{m2}$ is sequentially operated by $\mathcal{T}_c(\cdot)$ twice, which helps accumulate both the target-to-source and source-to-target information. After that, the enhanced feature $\tilde{F}_{m2}$ can be more powerful and representative for cost-volume building and initial flow producing.

After the spatial relation learning and mutual dependencies learning with OFE, the enhanced features of the two images are representative for cross-image matching. Therefore, we can apply the Softmax-based feature matching [36] to obtain the coarse flow. Since we intend to reduce the computational complexity of the recurrent decoding process, the truncated light decoder loses the original capability to handle the long-range regression problem. Therefore, the produced global motion should be reliable and coarsely match to the nearby region of the target. However, the general single-scale feature matching may result in large matching errors due to improper long-range affinities. We illustrate a typical failure case of feature matching in Fig. 4. As we can see, in scale $1/16$, the matching points locate at the reasonable positions (*i.e.* on the same car). However, in the generally used feature map at scale $1/8$, the peak activation appears in another car, leading to an unreliable global matching result.

To alleviate this issue, we design a multi-scale feature matching module. Specifically, given the enhanced feature $\tilde{F}_{m1}, \tilde{F}_{m2} \in \mathbb{R}^{h\times w\times d_m}$ from OFE and contextual feature $F_c \in \mathbb{R}^{h\times w\times d_c}$ from contextual encoder, we formulate the multi-scale feature matching function $\mathcal{M}(\cdot)$ as

$$\mathcal{M}(\tilde{F}_{m1}, \tilde{F}_{m2}, S) = \mathcal{O}(\frac{\tilde{F}_{m1} \cdot \tilde{F}_{m2}^{S\top}}{\sqrt{d_m}}) \cdot G^S \times S - G, \tag{2}$$

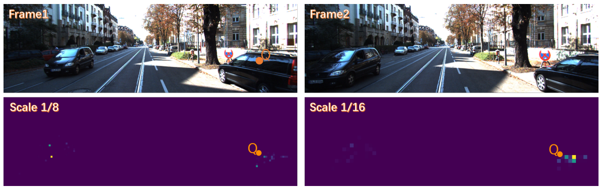


Figure 4: Visualisation of correlation maps in KITTI. "Q" denotes the query point in frame 1, and the activations of feature similarity in frame 2 are provided in the second row.

where $\mathcal{O}(\cdot)$ means softmax function. $G \in \mathbb{Z}^{h\times w\times 2}$ denotes 2D-grid coordinates, and S indicates the downsampling scale factor. Specifically, the produced flow maps are denoted as f_s, and we set $S \in \{1, 2, 4\}$, *i.e.*, obtaining global flow maps in scale $1/8, 1/16$ and $1/32$ of input size. After that, we design a flow fusion function $\mathcal{F}(\cdot)$ to combine the produced multi-scale flows, which is given by

$$\mathcal{F}(\mathrm{f}) = \mathcal{C}_1([\mathcal{C}_d(\mathrm{f})]), \tag{3}$$

where $\mathrm{f} = [\mathrm{f}_s]$, and $[\cdot]$ represents the concatenating operation along coordinate dimension or feature dimension, $\mathcal{C}_d(\cdot)$ represents convolution with dilation d, and s represents the scale. Specifically, we set $d \in \{4, 8, 16\}$, $s \in \{2, 4\}$, and the combination of multi-scale flows from $\mathcal{F}(\cdot)$ is denoted as f_c. To help initial flow estimation, we fuse f_c into $1/8$ scale flow by a weighting function $\mathcal{W}(\cdot)$, which is given by

$$\mathcal{W}(\mathrm{f}_{s=1}, \mathrm{f}_c) = \mathrm{f}_{s=1} + \gamma\, \mathrm{f}_c, \tag{4}$$

where γ is a learnable parameter. The output of $\mathcal{W}(\cdot)$ is denoted as f_i, representing the initial flow generated by MMA.

3.2. Confidence-induced Flow Propagation

With the multi-scale feature matching module, we can alleviate the unreliable matching in some degree. In light of the severe effect arising from erroneous global matching, the incorporation of an additional module to ensure the reliability of the matching process becomes imperative. Recently, GMFlow [36] proposes to use flow propagation to aggregate the flow estimation via the related context feature. However, this method is possible to aggregate some unreliable matching points, which are useless or even have a negative effect on flow estimation. To solve this problem, we propose to evaluate the matching confidence before aggregation. We empirically illustrate that the correlation between each frame pair can be utilized to generate confidence maps, as shown in Fig. 5. The confidence maps in the third row are obtained by selecting the maximum results within frame2's dimension of the 4D correlation. As we can see, a distinct correspondence can be observed between the error map and the confidence map, where higher errors are typically associated with lower confidence levels.

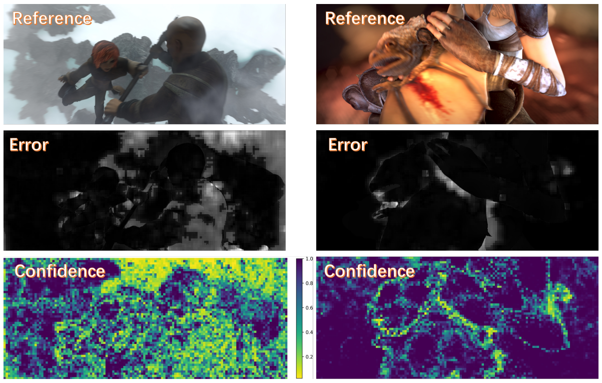


Figure 5: Confidence and EPE maps of the reference frames on Sintel final pass. Larger errors typically occur in regions of lower confidence.

Therefore, we propose to use confidence to induce the flow propagation. Specifically, given feature $\tilde{F}_{m1}$, $\tilde{F}_{m2}$ and downsampling scale factor S, we generate the confidence maps with function $\mathcal{G}(\cdot)$:

$$\mathcal{G}(\tilde{F}_{m1}, \tilde{F}_{m2}, S) = \max_{u,v}(\text{softmax}(\frac{\tilde{F}_{m1} \cdot \tilde{F}_{m2}^{S\top}}{\sqrt{d_m}})), \quad (5)$$

where u, v represent the height and width of $\tilde{F}_{m2}^{S}$, respectively. We denote $c_s = \mathcal{G}(\cdot)$ so that $c_s \in \mathbb{R}^{h\times w}$.

Before using c_s to induce propagation, we consider a situation that the feature matching between $\tilde{F}_{m1}$ and $\tilde{F}_{m2}^{S}$ may be ambiguous in a small region due to the similar context, which makes the peak value of confidence maps not obvious after $\mathrm{softmax}$ function. Therefore, we set a threshold α for confidence maps. When the confidence is larger than α, we think the confidence is high enough so that the flow estimation at that point can be regarded as reliable. Using α, the confidence maps can be represented as

$$c_r = \left\{ \begin{array}{cc} 1.0 & c_s > \alpha \\ c_s & c_s <= \alpha \end{array} \right. . \quad (6)$$

After obtaining c_r, we can leverage the reliability of flow estimating at each point. Additionally, in flow propagation, we avoid utilizing the points with low confidence. Therefore, we design a mask to guide the propagation. The mask m is defined as

$$\mathrm{m} = \left\{ \begin{array}{cc} 0 & \mathrm{c}_s > \alpha \\ -100 & \mathrm{c}_s <= \alpha \end{array} \right. , \quad (7)$$

where $\mathrm{m} \in \mathbb{R}^{h\times w}$. For all of the points in $\tilde{F}_{m1}$, the unreliable maps should be the same, so we can easily expand the dimension of m to $\mathbb{R}^{h\times w\times h\times w}$ with repeating operation.

Given the context feature F_c, mask m and the flow f_s, the flow propagation function $\mathcal{P}(\cdot)$ can be defined as

$$\mathcal{P}(F_c, \mathrm{m}, \mathrm{f}_s) = \text{Softmax}(\frac{F_c \cdot F_c^{\top}}{\sqrt{d_c}} + \mathrm{m}) \cdot \mathrm{f}_s. \quad (8)$$

The flow generated by $\mathcal{P}(\cdot)$ is denoted as $\hat{\mathrm{f}}_s$. After that, the confidence-induced function $\mathcal{I}(\cdot)$ can be formulated as

$$\mathcal{I}(\mathrm{f}_s, \hat{\mathrm{f}}_s, \mathrm{c}_r) = c_r \times \mathrm{f}_s + (1 - c_r) \times \hat{\mathrm{f}}_s. \quad (9)$$

3.3. Cascade Refinement

Presented so far, the architecture has already been able to achieve competitive performance and running efficiency (see EMD-S in Tab. 1). To further improve the performance, we follow GMFlow [36] to employ a cascade refinement module on 1/4 resolution features, namely EMD-M. Specifically, we reuse the encoder and change the last stride from 2 to 1, obtaining the basic motion feature $F_{m1}^{1/4}, F_{m2}^{1/4}$ and context feature $F_c^{1/4}$. Then, $F_{m1}^{1/4}$ and $F_{m2}^{1/4}$ are used to build a single scale 4D cost-volume. Furthermore, the initial hidden state $h_0^{1/4}$ and the initial flow $\mathrm{f}_i^{1/4}$ of the update block are obtained by performing bilinear interpolation on h_t and f_t, which are the final hidden state and cumulative flow in 1/8 stage, respectively. Finally, given the 4D cost-volume, $F_c^{1/4}$, $\mathrm{h}_0^{1/4}$ and $\mathrm{f}_i^{1/4}$, we can build a recurrent update scheme on 1/4 resolution to further refine the flow.

3.4. Network Instantiation

Following the prior works[32, 18, 38], we develop our EMD-Flow based on RAFT structure. Specifically, we first extract feature F_{m1}, F_{m2} by three residual layers with stride 2 and the feature dimension is d_m. The context feature F_c is extracted by three similar residual layers but with a lower dimension, and the output dimension is d_c. The feature F_{m1}, F_{m2} and F_c are fed to MMA and CFP to estimate the reliable initial flow. Then we construct the 4D correlation volume in one scale for all pairs of pixels and refine the flow by recurrent blocks.

In recurrent block, a motion encoder is designed to capture motion features by matching costs from 4D correlation volume and current optical flow. In the first recurrent step, the flow is f_i and the matching costs also correspond to f_i. It is worth noting that with the help of MMA and CFP, recurrent steps are mainly responsible to refine small displacement, so the recurrent unit is lightweight in EMD-Flow.

4. Experiments

Datasets and Training Schedule. Following prior works, we first pretrain our model on FlyingChairs[9] for 100k iterations with batch size 10 and then on FlyingThings[24] for 200k iterations with a batch size of 6. After that, we finetune the model on a combination of FlyingThings, Sintel[7], KITTI-2015[25] and HD1K[19] for 160k iterations, and the flow predictions are submitted to Sintel server[7] for online testing. Finally, to evaluate the performance on KITTI-2015 benchmark, additional 50K iterations are performed

Method	Iteration	Sintel(train, clean)				Sintel(train, final)				KITTI-15(train)					Param	Time
		EPE	s_{0-10}	s_{10-40}	s_{40+}	EPE	s_{0-10}	s_{10-40}	s_{40+}	EPE	F1-all	s_{0-10}	s_{10-40}	s_{40+}	(M)	(s)
RAFT[32]	1	4.04	0.77	4.3	26.65	5.45	0.99	6.3	35.18	15.3	44.5	1.13	3.94	30.44	5.3	0.008
	4	1.92	0.47	2.32	11.37	3.25	0.65	4.00	20.03	7.84	24.1	0.78	2.10	14.94		0.016
	12	1.53	0.38	1.71	9.25	2.81	0.53	3.30	17.92	5.41	18.1	0.71	1.67	9.97		0.038
	32	1.46	0.36	1.65	8.87	2.67	0.51	2.96	17.32	5.00	17.4	0.70	1.68	9.16		0.094
EMD-S	1	2.52	0.63	2.72	15.48	4.10	0.97	4.67	25.03	18.04	42.3	1.17	4.17	36.83	4.5	0.024
	5	1.44	0.30	1.49	9.41	2.92	0.60	3.35	18.43	9.27	24.1	0.73	2.07	17.67		0.028
	18	**1.31**	0.28	1.34	8.54	**2.67**	0.51	2.92	17.5	**5.00**	**17.0**	0.69	1.76	9.32		0.057
EMD-M	1	2.73	0.72	3.08	16.24	4.30	1.03	5.04	25.85	18.07	42.8	1.21	4.42	36.67	4.7	0.022
	3	1.42	0.34	1.71	8.46	2.99	0.66	3.54	18.27	12.44	29.7	0.81	2.55	24.27		0.028
	18	1.18	0.21	1.12	8.20	2.64	0.50	2.81	17.45	4.96	14.6	0.58	1.53	9.33		0.060
	24	**1.18**	**0.21**	**1.10**	**8.20**	**2.60**	**0.47**	**2.73**	**17.38**	**4.50**	**14.1**	**0.58**	**1.51**	**8.47**		0.074

Table 1: **Comparison between the pure recurrent process in RAFT and our EMD-Flow.** All models are trained on FlyingChairs[9] and FlyingThings[24] training set. The model and pretrained weight of RAFT are obtained from the official website. The inference time is measured with a single NVIDIA A100 card on Sintel dataset. **Bold** and underline denote the best results and the score surpassing RAFT, respectively.

Method	Params	Time	Sintel(train)		KITTI-15(train)	
	(M)	(s)	Clean	Final	EPE	F1-all
RAFT [32]	5.3	0.094	1.43	2.71	5.04	17.4
GMA [18]	5.9	0.130	1.30	2.74	4.69	17.1
AGFlow [23]	5.6	0.105	1.31	2.69	4.82	17.0
GMFlow [36]	4.7	0.072	1.08	2.48	7.77	23.4
KPAFlow [21]	5.8	0.159	1.28	2.68	4.46	15.9
CRAFT [29]	6.3	0.254	1.27	2.79	4.88	17.5
EMD-S(ours)	**4.5**	**0.057**	1.31	2.67	5.00	17.0
EMD-M(ours)	**4.7**	**0.074**	1.18	2.60	4.50	14.1
GMFlowNet [38]	9.3	0.137	1.14	2.71	4.24	15.4
FlowFormer [12]	18.2	0.930	0.95	**2.35**	**4.09**	14.7
EMD-L(ours)	13.7	0.120	**0.88**	2.55	4.12	**13.5**

Table 2: Comparison of parameter quantity, efficiency, and performance among recent SOTA methods. The results on Sintel [7] and KITTI-15 [25] are obtained from the models trained on FlyintChairs [9] and FlyingThings [24].

on KITTI-2015 training set. During training, AdamW optimizer with one-cycle learning rate scheduler is applied.

Implementation details. The EMD-Flow and experiments are implemented based on Pytorch. In both EMD-S and EMD-M, we set the dimensions of feature d_m and d_c to 128 and 64, respectively. In EMD-L, these dimensions are increased to 256 and 128. There are 2 OFE blocks in MMA and the scale number is set to 3. In CFP module, the confidence threshold α is configured as 0.4.

4.1. Comparison with RAFT

To evaluate the generalization ability of our EMD-Flow, we train the model on FlyingChairs and FlyingThings, and evaluate the performance on Sintel and KITTI.

Results on Sintel. As shown in Tab. 1, our EMD-S (with only 5 decoding iterations) achieves an equivalent EPE score with RAFT on Sintel clean only consuming $1/3$ runtime of RAFT and the final result surpasses RAFT by 10.3% (1.46 → 1.31). Besides, EMD-M model is more powerful, outstripping RAFT by 19.2% (1.46 → 1.18) still with runtime superiority. On the final pass, EMD-M outperforms RAFT by 2.6% (2.67 → 2.60). In addition, benefitting from the proposed motion disentangling module, our approach needs fewer iterations to obtain the equivalent performance (marked with underline).

Results on KITTI. Our EMD-S achieve the results of 5.00 in EPE and 17.0 in F1-all, which is better than RAFT. Moreover, EMD-M greatly surpasses RAFT by 10% (5.00 → 4.50) in EPE, 19.0% (17.4 → 14.1) in F1-all score.

4.2. Comparison with State-of-the-Arts

We compare our EMD-Flow with recent SOTA models in Tab. 2. Based on the number of parameters, recent approaches can be classified into two categories: small models and (relatively) large models. In the group of small models, EMD-S has superiority in parameter quantity and runtime. It still achieves competitive performance on both datasets. EMD-M is also efficient and outperforms most methods on Sintel, except GMFlow [36]. But GMFlow doesn't perform well on KITTI-15, while our EMD-M achieves the best F1-all score and ranks 2nd on EPE metric, just slightly falling behind KPAFlow [21], which contains 5.8 M parameters and consumes 0.206 seconds per frame. In another group, compared with the SOTA method FlowFormer [12], EMD-L achieves 7.4% (0.95 → 0.88) and 8.2% (14.7 → 13.5) improvement on Sintel clean pass and KITTI-15 F1-all, respectively, which set new state-of-the-art records on the two validation sets. Additionally, it is worth noting that EMD-L is also efficient, saving 85.0% (0.964 → 0.145) runtime compared to FlowFormer. More intuitive comparisons are depicted in Fig. 1 and Fig. 6.

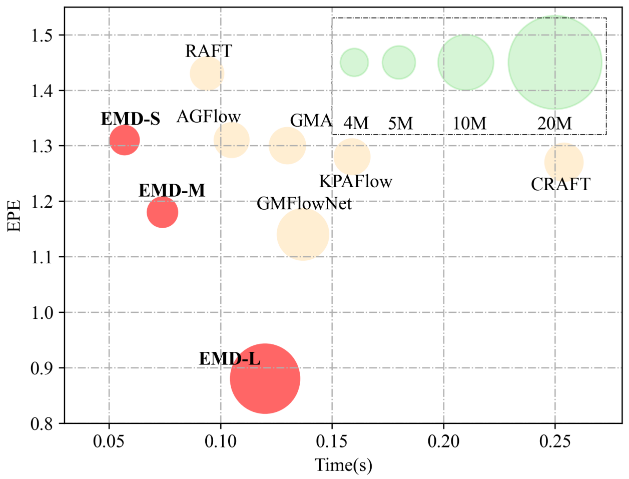


Figure 6: Comparison with state-of-the-art methods in terms of inference accuracy (EPE), runtime (s) and model size (M). All models are evaluated on Sintel clean with a single NVIDIA A100 card.

Method	Sintel(test)		KITTI-15(test)	Time (s)
	Clean	Final	F1-all	
AGFlow [23]	1.43*	2.47*	4.89	0.163
GMA [18]	1.39*	2.47*	5.15	0.146
KPAFlow [21]	1.35*	2.36*	4.60	0.206
CRAFT [29]	1.45*	2.42*	4.79	0.254
RAFT [32]	1.94	3.18	5.10	0.094
GMFlow [36]	1.74	2.90	9.32	**0.069**
GMFlowNet [38]	1.39	2.65	4.79	0.137
FlowFormer [12]	**1.14**	**2.18**	4.68	0.930
EMD-L(ours)	1.32	2.51	**4.51**	0.120

Table 3: Comparison on Sintel [7] and KITTI-15 [25] online benchmarks. * denotes using warm-start strategy [32].

4.3. Comparison on Benchmarks

Results on Sintel. The results on Sintel test set are shown in Tab. 3. We utilize the two-view setting and submit the predicted results to the official server of Sintel for online testing. Our EMD-L achieves EPE scores of 1.32 and 2.51 on Sintel clean and final pass, respectively. Under the setting of two-view, EMD-L ranks 2nd on Sintel benchmark, only falling behind FlowFormer [12]. However, compared with FlowFormer, EMD-L saves 24.7% parameters (18.2M → 13.7M) and 85.0% running time (0.964s → 0.145s).

Results on KITTI. As shown in Tab. 3, our EMD-L achieves an F1-all score of 4.51, ranking 1st on the KITTI-15 benchmark, which outperforms top-ranked methods FlowFormer [12] and GMFlowNet [38] by 3.6% (4.68 → 4.51) and 5.8% (4.79 → 4.51), respectively.

Method	Sintel(train)		KITTI-15(train)		Δ
	clean	final	EPE	F1-all	
RAFT (single scale)	1.70	2.81	6.01	19.6	-
+ Soft. & Prop. [36]	1.59	2.83	6.78	20.3	- 2.7%
+ Arg. [38]	1.43	2.83	6.56	20.2	+ 0.7%
+ Arg. [38] & Prop. [36]	1.56	2.78	6.19	18.4	+ 3.1%
+ CSA [35] & CFP	1.84	2.97	7.07	21.9	-11.7%
+ MMA & Prop. (ours)	1.43	2.77	5.86	18.7	+ 6.1%
+ MMA & CFP (ours)	1.36	2.79	5.74	18.1	+ 8.2%

Table 4: Ablation experiments. All of the models are trained on FlyingChairs [9] and FlyingThings [24] for 100K iterations, respectively. Δ represents average an improvement compared with RAFT (single scale).

Multi Scale	Sintel(train)		KITTI-15(train)		Params
	clean	final	EPE	F1-all	(M)
w/.	1.18	**2.60**	**4.50**	**14.1**	4.7
w/o	**1.15**	2.64	4.74	14.8	4.5

Table 5: Ablation for multi-scale in MMA module.

4.4. Ablations

Ablation for Multi-scale Motion Aggregation. We compare our MMA module with softmax method [36] and argmax method [38]. As can be seen in Tab. 4, with the same flow propagation [36], our MMA module has 6.1% average improvement, higher than -2.7% of softmax method and 3.1% of argmax method. To further validate the effect of multi-scale structure in MMA, we also ablate the multi-scale structure in the fully trained EMD-M, as shown in Tab. 5. The performance is improved greatly by 5.1%(4.74 → 4.50) on KITTI EPE and 4.7%(14.8 → 14.1) on KITTI F1-all. Some qualitative results are illustrated in Fig. 4 and we can find that due to the similar texture of two black cars, the 1/8 scale appears wrong activation of feature similarity in frame2, but the 1/16 scale can remedy this error in some degree. In addition, our MMA performs multi-scale flow aggregation, while CSA module of AANet [35] conducts the aggregation on cost that inevitably requires numerous parameters due to the large 4D cost-volume in optical flow. As shown in Tab. 4, we replace MMA with CSA for a fair comparison. The empirical results further demonstrate the superiority of our approach.

Ablation for Confidence-induced Flow Propagation. In addition, we also compare our CFP with flow propagation [36]. For a fair comparison, we just replace the flow propagation with our CFP, and other modules remain the same. In Tab. 4, compared with 6.4% average improvement of flow propagation, our CFP achieves 9.1% average improvement, which demonstrates the effectiveness of CFP.

Ablation for Cascade Refinement and Stronger Encoder.

	Setting	Sintel(train)		KITTI-15(train)		Params
		clean	final	EPE	F1-all	(M)
Baseline	EMD-S	1.31	2.67	5.00	17.0	4.5
CR	12+6 iters.	1.18	2.63	4.96	14.6	4.7
	18+6 iters.	1.18	2.60	4.50	14.1	4.7
SE	Twins [12]	0.88	2.55	4.12	13.5	13.7
	w/o.	1.18	2.60	4.50	14.1	4.7

Table 6: Ablation experiments. 'CR' and 'SE' denote Cascade Refinement and Strong Encoder, corresponding to EMD-M and EMD-L, respectively. All of models are trained on FlyingChairs [9] and FlyingThings [24].

Model	Sintel(train)		KITTI-15(train)		Δ	Time
	clean	final	EPE	F1-all		(s)
RAFT [32]	1.43	2.71	5.04	17.4	-	0.116
EMD-M(ours)	**1.18**	**2.60**	**4.50**	**14.1**	+ 12.9%	**0.098**
RAFT-GMA [18]	1.30	2.74	**4.69**	17.1	-	0.152
EMD-GMA(ours)	**1.01**	**2.43**	5.03	**16.0**	+ 8.2%	**0.141**
RAFT-KPA [21]	1.28	2.68	4.46	15.9	-	0.206
EMD-KPA(ours)	**1.06**	**2.55**	**4.42**	**14.1**	+ 8.5%	**0.202**
RAFT-Twins [32]	1.25	2.84	4.55	15.8	-	**0.113**
EMD-Twins(ours)	**0.88**	**2.55**	**4.12**	**13.5**	+ 16.0%	0.145

Table 7: The results of a combination between our EMD-M and recent works. Δ represents the average improvement brought by EMD-Flow.

EMD-S is a high-efficiency basic structure, which can surpass RAFT [32] with less time cost. To further improve the performance of EMD-Flow, we propose EMD-M and EMD-L, and the comparison among three EMD-Flow models is shown in Tab. 6. With low parameters growth, the cascade refinement can greatly improve the performance by 10%(1.31 → 1.18) on Sintel clean, 10% (5.00 → 4.50) on KITTI EPE and 17%(17.0 → 14.1) on KITTI F1-all score. It is worth noting that although using cascade refinement, our EMD-M model still has higher efficiency than RAFT. Similar to [12], we also try to substitute the convolution encoder with Transformer encoder to further enhance the performance of EMD-Flow. Seeing Tab. 6, with stronger encoder Twins[12], EMD-Flow obtains better results on both Sintel and KITTI.

Combination with recent methods. EMD-Flow is compatible with recent advanced methods and has great potential to serve as a strong baseline. In Tab. 7, we provide the results of the combination between EMD-Flow and recent advanced methods. EMD-GMA is the combination of our EMD-M and Global Motion Aggregation (GMA) module [18]. We calculate the self-attention on the query and key vectors embedded from the context feature. Then the value vector from the encoded matching cost is involved to aggregate motion features before the GRU update module. Besides, we also employ Kernel Patch Attention (KPA) [21] in our model. Similar to the process in EMD-GMA, we apply the kernel-based function on the context embeddings to obtain attention maps, and then perform the regional motion refinement. This model is termed EMD-KPA in Tab. 7. For developing EMD-Twins, we simply replace the ResNet encoder used in EMD-M model with the Twins Transformer as in FlowFormer [12]. Compared with the combination between RAFT and advanced methods, EMD-Flow can obtain up to 8.2% to 16.0% improvement with very low or even no extra overhead.

Method	Params	Time	Sintel(train)		KITTI-15(train)	
	(M)	(s)	Clean	Final	EPE	F1-all
MS-RAFT [†]	13.5	0.238	1.13	2.60	-	-
EMD-M [†]	4.7	0.098	1.12	2.52	-	-
EMD-M	4.7	0.098	1.18	2.60	4.50	14.1

Table 8: Additional comparisons. [†] indicates using the warm-start strategy [17] for fair comparison.

Comparison with MS-RAFT. The main differences between the proposed MMA and MS-RAFT can be summarized in two-fold: First, their motivations are completely distinct. Our MMA aims to enhance the robustness of global matching by using additional compressed features. In contrast, MS-RAFT is intended to elevate flow accuracy for fine-grained details by utilizing more high-resolution features. Second, MMA is highly efficient and demands minimal computational resources, for it is implemented using the 1/8 scale features and the further pooled ones. Conversely, MS-RAFT employs a hierarchical architecture on features across $\{1/16,\ 1/8,\ 1/4\}$ scales, leading to significant computational overhead, as shown in Tab. 8.

5. Conclusion

We propose EMD-Flow, a novel architecture that explicitly disentangles the motion and accelerates the recurrent estimating process of RAFT. Our approach enables efficient flow estimation by using lightweight methods for both parts and can be further improved with the MMA and CFP modules. We demonstrate that EMD-Flow achieves a new state-of-the-art performance in terms of the balance between accuracy and efficiency on both the Sintel and KITTI benchmarks. We expect that our work will provide a new perspective in improving optical flow estimation tasks.

Acknowledgements. This work was supported in part by Sichuan Science and Technology Program of China under grant No.2023NSFSC0462, in part by the Science and Technology Development Fund, Macau SAR under Grants 0087/2022/AFJ and SKL-IOTSC(UM)-2021-2023, in part by the National Natural Science Foundation of China under Grant 62261160650.

References

[1] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015.

[2] Michael J Black and Padmanabhan Anandan. A framework for the robust estimation of optical flow. In *ICCV*, 1993.

[3] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 1996.

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[5] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *TPAMI*, 2010.

[6] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Combining the advantages of local and global optic flow methods. In *Joint Pattern Recognition Symposium*, 2002.

[7] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[8] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Understanding deformable alignment in video super-resolution. In *AAAI*, 2021.

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[10] Yunhui Han, Kunming Luo, Ao Luo, Jiangyu Liu, Haoqiang Fan, Guiming Luo, and Shuaicheng Liu. Realflow: Em-based realistic optical flow dataset generation from videos. In *ECCV*, 2022.

[11] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 1981.

[12] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A transformer architecture for optical flow. In *ECCV*, 2022.

[13] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 2022.

[14] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018.

[15] Junhwa Hur and S. Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 2019.

[16] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[17] Azin Jahedi, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. Multi-scale RAFT: Combining hierarchical concepts for learning-based optical flow estimation. In *ICIP*, 2022.

[18] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021.

[19] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *CVPRW*, 2016.

[20] Shuaicheng Liu, Kunming Luo, Ao Luo, Chuan Wang, Fanman Meng, and Bing Zeng. Asflow: Unsupervised optical flow learning with adaptive pyramid sampling. *TCSVT*, 2021.

[21] Ao Luo, Fan Yang, Xin Li, and Shuaicheng Liu. Learning optical flow with kernel patch attention. In *CVPR*, 2022.

[22] Ao Luo, Fan Yang, Xin Li, Lang Nie, Chunyu Lin, Haoqiang Fan, and Shuaicheng Liu. Gaflow: Incorporating gaussian attention into optical flow. In *ICCV*, 2023.

[23] Ao Luo, Fan Yang, Kunming Luo, Xin Li, Haoqiang Fan, and Shuaicheng Liu. Learning optical flow with adaptive graph reasoning. In *AAAI*, 2022.

[24] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[25] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[26] AJ Piergiovanni and Michael S Ryoo. Representation flow for action recognition. In *CVPR*, 2019.

[27] René Ranftl, Kristian Bredies, and Thomas Pock. Non-local total generalized variation for optical flow estimation. In *ECCV*, 2014.

[28] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017.

[29] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformer for robust optical flow. In *CVPR*, 2022.

[30] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[31] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, 2021.

[32] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[33] Mikko Vihlman and Arto Visala. Optical flow in deep visual tracking. In *AAAI*, 2020.

[34] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

[35] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *CVPR*, 2020.

[36] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022.

[37] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, 2007.

[38] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *CVPR*, 2022.